

Preface

Software application development is moving closer to a long-held goal: engineering new systems by manipulating existing software parts and templates, writing new software only when necessary to add or change function or to glue together other parts. This issue is on methods and tools being used at IBM and elsewhere to approach that goal of application development through a software engineering style and discipline.

The issue contains an introductory essay on application development as an engineering discipline, eight papers on aspects of modern software development and studies of results, and a Technical Forum on CommercePOINT*. We are indebted to N. Bieberstein of the IBM Arthur K. Watson International Education Center in La Hulpe, Belgium, for his significant contributions to the planning, coordination, and development of this issue.

The nature of software development has slowly but surely evolved toward a state in which engineering principles and an engineering approach prevail. Application development has made significant strides in this direction. Bieberstein, in an introductory essay, describes aspects of this historical path and the desire to duplicate the successes of other engineering fields, while acknowledging what is unique about software.

Prins, Blokdijk, and Van Oosterom describe a method of software development focused on the needs and styles of businesses. The number and complexity of objects needed for business applications require methods of software development that start with an understanding of business structures and the advantages of software reuse, combined with the need for a business advantage through rapid application development. The resulting approach has been borne out by project experiences, which the authors illuminate with lessons learned.

Any fundamentally new approach to product development requires significant effort from those involved with product creation; it is difficult to move from one style to another. Object technologies for software development suffer from this same phenomenon. Lindsey and Hoffman address this aspect of technology evolution by presenting two methods for easing the transition for object technologies: bridging between the old and new technologies, and masking the difficulties in adopting the new technologies.

The challenge of interoperability between object-oriented applications and existing non-object-oriented applications must be resolved at various levels within software development. Flint considers the concept and problems of object wrappers in COBOL and then provides an approach that expands the applicability of the wrapper approach. The author adds procedural and combination wrappers to the tools available for handling interoperability in COBOL.

Srinivasan and Chang address the basic question of object persistence and its impact on the storage of data by object-oriented and non-object-oriented applications, which is another example of the challenge of interoperability. Their paper describes three mechanisms for handling persistent data: gateway-based object persistence, object-relational database management systems, and object-oriented database management systems. The advantages and disadvantages of each are explored for various application situations.

Debugging user-developed software running on a server is made difficult by the inability of most debugging tools to penetrate the system-provided software that is provided without source code. Meier, Pan, and Fuh demonstrate one way in which such difficulties can be overcome by prototyping and testing a set of extensions to a distributed debugger and Database 2*/Common Server (DB2*/CS), without jeopardizing DB2/CS security.

Workflow-based applications, as reported by Leymann and Roller, can provide solutions for businesses that are re-engineering their processes and their associated software systems for the distributed, heterogeneous, networked environments of today and tomorrow. The authors follow a path that derives applications from business process models, using workflow to provide structure for process definition and software development.

A major study has been completed on the effectiveness of tools used for application development. That study, conducted by and reported on by Guinan, Coopridier, and Sawyer, covered over 100 projects carried out by major companies. It measured the correspondence between use of tools at the application development level and user-sensitive values, such as usability and timeliness. The study shows that the successful use of tools is a combination of many factors, that the tools themselves are only one factor, and that the results of tool use are mixed.

Potok and Vouk explore another aspect of the impact of methods on software results: the effect of business models on software development productivity in an object-oriented environment. The thesis is that software development is driven by business models and not the other way around. The authors question whether the object-oriented software development paradigm enhances productivity and, using commercial empirical data, conclude that it may or may not, depending on incentives reflected in the business model.

The final contribution, by Connolly, appears in the Technical Forum. The author describes the Internet-based technologies used in IBM's CommercePOINT, an electronic commerce system for both business-to-business and business-to-consumer applications. This system provides a wide range of access capabilities through the Internet, with the secure infrastructure and 24-hour operation required for commercial acceptability worldwide.

As the *Journal* begins its 36th year, we would like to acknowledge the support of readers, authors, and referees that makes such a long history possible. We thank you and encourage you to continue your interest and participation in this publication. In 1996, the 85 authors represented IBM (32 percent) and other sources (68 percent), and the United States (89 percent) and other countries (11 percent). It also seems appropriate at such a time to state a few facts that sometimes escape us as we focus on a single pa-

per, theme, or special issue. First, this publication is a quarterly refereed technical journal, which means that the integrity of each paper is ensured by a process that depends upon peer reviews of content, currency, and value by recognized experts within and outside IBM. Second, it is intended for the software and systems professional and applied research community worldwide. The papers are written for a technically aware readership, and we welcome submissions by knowledgeable authors around the globe, within and outside IBM. Third, the *Journal* has over 55 000 subscribers worldwide. Of those, approximately two-thirds are technical professionals and researchers outside IBM; one-third are IBM employees; two-thirds are in the United States; and one-third are in other countries.

The next issue of the Journal will be a special issue on IBM System/390* Parallel Sysplex*, a multisystem parallel processing environment.

Gene F. Hoffnagle
Editor

*Trademark or registered trademark of International Business Machines Corporation.